




[hash] Create a hash table - dò bậc 2 (using array

 [Download the code template](#)

MÔ TẢ BÀI TOÁN

Hãy viết chương trình thực hiện thao tác “Nhập xuất cơ bản một bảng băm chứa các số nguyên” với những yêu cầu sau:

- Giải quyết xung đột bằng phương pháp DÒ BẬC 2. Thường cài đặt bảng băm bằng ARRAY
- Nếu băm bị xung đột thì băm lại. Quá trình băm lại diễn ra cho đến khi không còn xung đột nữa. Các phép băm lại thường sẽ chọn địa chỉ khác cho các phần tử.

- PHƯƠNG PHÁP DÒ BẬC HAI

Hàm băm lại của phương pháp dò bậc hai là truy xuất các địa chỉ cách bậc 2

$f_j(\text{key}) = (f(\text{key}) + j^2) \% M$, với $f(\text{key})$ là hàm băm chính của bảng băm

- + Khi thêm 1 nút có khóa key , $f(\text{key})$ sẽ xác định địa chỉ i :
- + Nếu chưa bị xung đột thì thêm nút mới tại địa chỉ i này.
- + Nếu bị xung đột thì băm lại lần 1, f_1 sẽ xét địa chỉ $(i + 1^2) \% M$,
- + Nếu lại bị xung đột thì băm lại lần 2, f_2 xét địa chỉ cách $(i + 2^2) \% M$, ...
- + Tiếp tục cho đến khi tìm được địa chỉ trống để thêm vào.
- + Khi bảng băm bị đầy thì xuất thông báo `cout << "\nBang bam bi day, khong them duoc";` Bảng băm đầy khi số lượng phần tử bằng số địa chỉ.

LƯU Ý: 1 GIẢI PHÁP TRIỆT ĐỂ CẦN PHẢI XỬ LÝ NHỮNG TRƯỜNG HỢP ĐẶC BIỆT SAU:

Vòng lặp băm lại bị vô tận hoặc số lần băm lại lớn quá dẫn đến $i \cdot i$ quá lớn và bị tràn số (vượt khỏi miền giá trị của `int`) hoặc không thêm được do không thỏa 1 điều kiện cho trước nào đó

1. Nhận biết băm lại vô tận/không thêm được: băm lại hoài mà chỉ trả về cùng 1 giá trị, hoặc là quay lại những giá trị trước đó (tức có 1 chu kỳ quanh đi quẩn lại 1 số giá trị nào đó), hoặc là đã thăm qua hết các địa chỉ hoặc dựa trên hệ số tải quy ước sẵn (bảng băm gần đầy) hoặc **băm lại M lần rồi mà vẫn không thêm được. Nhiều tài liệu tham khảo ghi là "hàm insert tiến hành probing và dừng khi $i = M$** .

2. Khi bị lặp vô tận/không thêm được thì chọn phương án xử lý

Cách 1: ghi error " không thêm được"

Cách 2-cách làm phổ biến nhất trong thực tế: tạo bảng băm mới, add lại hết các số cũ sau đó xóa bảng cũ đi, xem như là thay đổi bảng băm



không cần xử lý ngoại lệ (trừ trường hợp bảng băm đầy), và không được các test.

VÒNG LẶP BẮM LẠI KHÔNG ĐẶT ĐIỀU KIỆN DỪNG $i == M$, cứ để chương trình băm hoài đến khi tìm được vị trí hoặc bị tràn số

INPUT

- Số nguyên đầu tiên là n ($0 \leq n \leq 10000$). Đây là số lượng phần tử sẽ được lưu trữ vào bảng băm

- n số nguyên tiếp theo là các (giá trị) khóa của các phần tử được lưu vào bảng băm, có giá trị không quá 2 tỷ.

**SV không cần kiểm tra điều kiện nhập, không quan tâm đến hệ số tải

OUTPUT

- Xuất theo template

- Khi bảng băm bị đầy thì xuất thông báo `cout<<"\nBang bam bi day, khong them duoc";` Bảng băm đầy khi số lượng phần tử bằng số địa chỉ.

VÍ DỤ

Input	Output	Giải thích
9		
44		
9		
27	Insert 44	
57	Insert 9	
36	Insert 27	
17	Insert 57	
22	Insert 36	
73	Insert 17	
91	Insert 22	
	Insert 73	
	Insert 91	
	Created Hash:	
	0	
	1 --> 17	
	2 --> 22	
	3 --> 73	



```
4 --> 44
5 --> 91
6 --> 36
7 --> 27
8 --> 57
9 --> 9
```

```
Insert 6
Insert 63
Insert 6
Insert 73
Insert 36
Insert 41
```

```
12 Insert 87
6 Insert 21
63 Insert 38
6 Insert 6
73 Insert 77
36 Bang bam bi day,
41 khong them duoc
```

```
87 Insert 71
21 Bang bam bi day,
38 khong them duoc
6 Created Hash:
```

```
77 0 --> 36
71 1 --> 41
    2 --> 21
    3 --> 63
    4 --> 73
    5 --> 6
    6 --> 6
    7 --> 6
    8 --> 87
    9 --> 38
```

```
9
46 Insert 46
82 Insert 82
```

VÒNG LẶP BẮM LẠI KHÔNG ĐẶT ĐIỀU KIỆN DỪNG $i == M$, cứ để chương trình bấm hoài đến khi tìm được vị trí hoặc bị tràn số



```

87  Insert 87          Insert 46: bam lai 0 lan
48  Insert 48          Insert 82: bam lai 0 lan
28  Insert 28          Insert 87: bam lai 0 lan
98  Insert 98          Insert 48: bam lai 0 lan
72  Insert 72          Insert 28: bam lai 1 lan
29  Insert 29          Insert 98: bam lai 4 lan
98  Insert 98          Insert 72: bam lai 1 lan
    Created Hash:      Insert 29: bam lai 1 lan
    0 --> 29           Insert 98: bam lai 65537 lan
    1 --> 98
    2 --> 82
    3 --> 72
    4 --> 98
    5
    6 --> 46
    7 --> 87
    8 --> 48
    9 --> 28

```

13. [Inclass#11] Hash Table (Pre-Intermediate)

18 problems with a total score of 1800

#	Problem	Score
1	[hash] Loại bỏ số trùng (dùng kỹ thuật băm dạng đơn giản nhất, ko có xung đột).	100
2	[hash] Tìm giá trị chỉ xuất hiện 1 lần trong ma trận (dùng kỹ thuật băm dạng đơn giản nhất, ko có xung đột).	100
3	[hash] Ký tự còn thiếu (dùng kỹ thuật băm dạng đơn giản nhất, ko có xung đột).	100
4	[hash] Find MEX (dùng STL.map).	100
5	[hash] Find MEX (dùng STL.set).	100



6	[hash] Find MEX (dùng STL.vector).	100
7	[hash] Kiểm kê (dùng STL).	100
8	[hash] Login 1 (dùng STL).	100
9	[hash] Create a hash table - dò tuyến tính (using array).	100
10	[hash] Create a hash table - dò bậc 2 (using array).	100
11	[hash] Game online	100
12	[hash] Login 2	100
13	[hash] Đền hoa	100
14	[hash] Detect virus	100
15	[hash] Bảng băm nối kết - Hàm Insert - 1 (gần giống với đề thi cũ).	100
16	[hash] Bảng băm nối kết - Hàm Search - 1 (gần giống với đề thi cũ).	100
17	[hash] Bảng băm thăm dò - Hàm Insert - 1 (gần giống với đề thi cũ).	100
18	[hash] Bảng băm thăm dò - Hàm Search - 1 (gần giống với đề thi cũ).	100